SOLAR: Serendipity Optimized Language Model Aligned for Recommendation

Anonymous ACL submission

Abstract

Recently, Large Language Models (LLMs) possess broad world knowledge and show potential in diversifying recommendations. They face two key challenges: a domain gap in capturing user behavior patterns and a scarcity of human-labeled data for serendipitous recommendations. In this paper, we propose **SOLAR**, a serendipity-optimized language model aligned for recommendation, which bridges these gaps through a three-step process. First, we train a ID-based model that balances accuracy and serendipity via human-centric labels. We then generate large-scale, high-quality fine-tuning data via a two-step prompting strategy using an LLM-based reranker. Finally, we construct a recommendation-specialized unified tuning network (**SUN**) to align the LLM with recommendation tasks using domain-adaptive instructions. Experiments across multiple datasets demonstrate that **SOLAR** consistently outperforms baseline models in both accuracy and serendipity, offering a promising solution to break free from filter bubbles and promote more diverse, user-centric recommendations.

1 Introduction

Recommender systems mainly rely on user historical behavior data to learn user interest patterns. Most recommendation algorithms are accuracy-driven, tending to recommend similar items to users in order to improve performance on accuracy metrics. However, these recommendation algorithms can lead to the creation of a filter bubble (Pariser, 2011). In other words, the system keeps recommending content similar to what users have already shown interest in, and users subsequently provide feedback on this limited set of items. This feedback loop reinforces the existing learning strategy and further narrows users' exposure to diverse content. As a result, the recommended items become more homogeneous, making it harder for users to discover new interests. Content fatigue eventually reduces their interest and engagement with the platform.

To address the filter bubble and enhance serendipity, researchers have explored various strategies. Previous studies (Pandey et al., 2018a; Chen et al., 2021; Ekstrand and Willemsen, 2016) have applied multi-objective optimization to balance accuracy with serendipity, aiming to improve overall recommendation quality. More recently, the emergence of large language models (LLMs) has opened new pathways to address these challenges. With their broad world knowledge, LLMs can overcome the limitations of traditional recommender systems that rely solely on domain-specific information. By directly exploring the recommendation space through the language space (Wang et al., 2024b; Sheng et al., 2024), LLMs can identify richer user interest patterns and inspire more diverse recommendations.

Despite these advantages, LLMs face two key challenges in practical applications. **(I) Domain gap:** Although LLMs excel in knowledge-based and reasoning tasks, they lack collaborative filtering capabilities, making it difficult to capture personalized user preferences from interaction data (Zhao et al., 2024; Ning et al., 2024; Lin et al., 2023). **(II) Label scarcity:** Serendipity-oriented recommendations are inherently subjective and require human ratings, which are typically scarce (Kotkov et al., 2018; Fu et al., 2023). Current evaluation methods often rely on assumptions rather than direct user feedback, defining serendipity through the latent distance between recommended items and user profiles (Li and Tuzhilin, 2020; Ge et al., 2010; Kotkov et al., 2016). These methods fail to incorporate real user feedback and thus do not accurately reflect human preferences.

To overcome the above challenges, we propose **SOLAR**, a serendipity-optimized language model aligned for recommendation. **SOLAR** involves three steps from a small set of available labels: First, we train a ID-based model that balances accuracy and serendipity; Second, we leverage LLM-based reranking to generate high-quality fine-tuning data at a large scale; Finally, we construct recommendation-oriented instruction sets to fine-tune the LLM for aligning with real human preferences. We adopted two metrics that take human annotations as the ground truth to validate the effectiveness of **SOLAR** in serendipity recommendation. Experiments show that our proposed LLM-based approach significantly outperforms baselines in both accuracy and serendipity.

The contributions of our work are summarized as follows:

1. We propose **SOLAR**, a serendipity-oriented LLM for recommender systems, which aligns with human preference to provide both accurate and novel recommendations.

2. **Serendipity-labeled data generation.** To mitigate the problem of label scarcity for serendipity data, we design a two-stage approach that combines the advantages of both serendipity-fine-tuned ID-based recommender models and LLM-based reranker for serendipity-labeled data generation.

3. **Domain-adaptive instruction paradigm.** We introduce a recommendation-specialized unified tuning network (**SUN**), representing a novel paradigm for domain adaptation. By encoding user history, engagement profiles, and interaction patterns into text-based instructions, we provide a flexible, scalable, and reusable framework that goes beyond conventional fine-tuning methods, effectively bridging the gap between general-purpose LLMs and recommendation tasks.

4. **Robust empirical validation.** We conduct comprehensive evaluations on multiple datasets, demonstrating that **SOLAR** outperforms existing state-of-the-art baselines in both accuracy and serendipity.

2 Related Works

2.1 LLMs for Recommender Systems

Recent advances in large language models (LLMs) have introduced new opportunities for enhancing recommender systems, extending their capabilities and application scenarios (Wang et al., 2023a; Zhao et al., 2023). Some works focus on generative recommendation methods (Geng et al., 2022; Li et al., 2023b; Wang et al., 2024b; Geng et al., 2023; Liu et al., 2023), while others use LLMs for feature engineering (Hou et al., 2023; Li et al., 2024) or for improving representation learning, thereby enriching both user and item embeddings (Ning et al., 2024; Xi et al., 2023; Li et al., 2023a; Zhang et al., 2023). To better align with real-world user habits, researchers have integrated LLMs into conversational recommendation scenarios (Yang et al., 2024; Gao et al., 2023; Feng et al., 2023) and LLM-powered agents (Zhang et al., 2024a,b; Xie et al., 2024) to simulate authentic user interactions.

Furthermore, incorporating complex and often noisy user interaction data remains a challenge in aligning LLM outputs with human preferences (Wang et al., 2024a; Ren et al., 2024). One approach to addressing this challenge involves combining LLMs with conventional recommendation models or fine-tuning them using domain-specific instructions, thereby ensuring more consistent responses (Li et al., 2023c; Wang et al., 2023b, 2024c). Initial explorations have examined using LLMs as rerankers, integrating them efficiently with existing recommendation pipelines (Hou et al., 2024; Pradeep et al., 2023; Ma et al., 2023; Yue et al., 2023a). These methods have shown that LLMs can deliver effective reranking performance without requiring extensive task-specific training data.

2.2 Serendipity Recommender Systems

In real-world recommender systems, it's crucial to balance accuracy with beyond-accuracy objectives, such as diversity (Díez et al., 2019; Rodriguez et al., 2012) and serendipity (Ge et al., 2010; Pardos and Jiang, 2020; Wang et al., 2024b) through multi-objective optimization. Learning from multiple metrics helps the system provide relevant recommendations while catering to user satisfaction, thus enhancing user engagement.

A key challenge lies in the closed-loop nature of traditional ID-based systems, as models learn from historical user-item interactions and struggle to uncover new user novel interests (Sepliarskaia et al., 2018; Rokach and Kisilevich, 2012). To overcome this limitation, LLMs, with their powerful semantic capabilities and world knowledge, offer a promising approach to breaking this feedback loop. They enable the modeling of more general and comprehensive user preference representations for recommendation (Gao et al., 2023; Luo et al., 2023; Zhao et al., 2024; Wang et al., 2023c) and facilitate the exploration of user interests (Li et al., 2023b). Building on this potential, our work introduces a novel approach that integrates world knowledge from LLMs and its ability to capture human subjective preferences, aiming to produce more diverse, serendipitous, and user-aligned recommendations.

3 Methodology

We propose a novel framework that combines ID-based sequential recommendation with LLMs to enhance recommendation quality by introducing serendipity. The framework consists of three steps: (1) Develop a hybrid recommendation model that integrates an ID-based sequential recommendation model with an LLM-based reranker. Fine-tune this model using limited, high-quality human-labeled data combined with LLM reranker to capture nuanced user interactions and better align recommendations with human preferences for serendipity. (2) Use the hybrid recommendation model to generate candidate lists. For each user, provide their history and the list to a powerful LLM. Prompt it to assess the serendipity of each recommendation from the user's perspective, generating explanations that will be used to construct an instruction for fine-tuning the LLM in the next step. (3) Create a comprehensive instruction set for fine-tuning the LLM on the task of generative recommendation based on the collected instruction data.

To begin, we formulate the sequential recommendation task. For each user u, we observe a historical interaction sequence $H = \{h_1^u, h_2^u, \ldots, h_n^u\}$, representing the items a user has interacted with up to time step n. The objective is to predict the next item h_{n+1}^u that the user is likely to interact with. Specifically, given

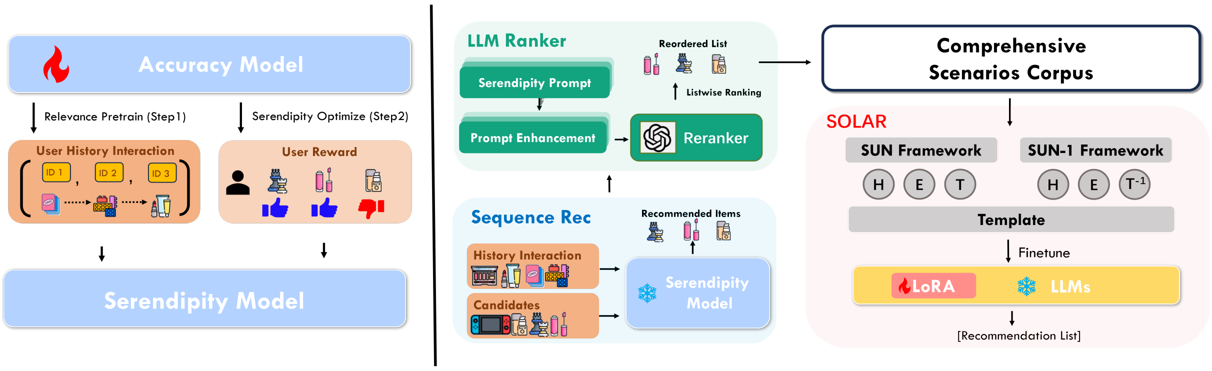


Figure 1: Framework of **SOLAR**: The framework consists of three key components: (1) a hybrid recommendation model by integrating an ID-based sequential recommendation model with an LLM-based reranker, and fine-tuning this model with serendipity loss (left and middle); (2) Generating instruction data by employing the hybrid model to produce candidate lists and prompting the LLM to assess the serendipity of each recommendation, providing explanations for its instructions (upper right); and (3) Constructing the Recommendation Specialized Unified Tuning Network (SUN) framework and fine-tuning the LLM using the generated instruction dataset to enhance serendipity-oriented recommendations (low right).

a user set $U = \{u_1, u_2, \ldots, u_{|U|}\}$ and an item set $I = \{i_1, i_2, \ldots, i_{|I|}\}$, our model estimates the conditional probability $\mathbb{P}(i \mid H, \boldsymbol{\theta})$ for each candidate item $i \in I$. The item predicted for the next interaction is then selected as:

$$h^u_{n+1} = \arg\max_{i \in I} \mathbb{P}(i \mid H, \boldsymbol{\theta}),$$

where $\boldsymbol{\theta}$ denotes the parameters of the model.

The following subsections elaborate on each step of the framework, including the fine-tuning of the ID-based model, the LLM-based reranking approach, and the construction strategy of instruction dataset.

3.1 ID-based Model Training

Consider a set of users U of size N and a set of items I of size M. Each user u is represented by its historical interaction sequence $H_u = \{i_{k_1}, i_{k_2}, \ldots, i_{k_l}\}$. Let $\boldsymbol{\theta}$ denote the model parameters. The model produces a prediction $p_{jk}(\boldsymbol{\theta})$ indicating the probability that the user u_j considers item i_k both relevant and serendipitous:

$$p_{jk}(\boldsymbol{\theta}) = p(i_k \mid u_j, \boldsymbol{\theta}).$$

We define two types of ground-truth label matrices: a relevance matrix $R = \{r_{jk}\}$ and a serendipity matrix $S = \{s_{jk}\}$. For a user u_j and an item i_k:

$$r_{jk} = \begin{cases} 1 & \text{if } u_j \text{ considers } i_k \text{ relevant} \\ 0 & \text{otherwise} \end{cases}$$

$$s_{jk} = \begin{cases} 1 & \text{if } u_j \text{ considers } i_k \text{ serendipitous} \\ 0 & \text{otherwise} \end{cases}$$

To jointly model relevance and serendipity, we minimize a combined loss function. Let $\mathbb{P}(\boldsymbol{\theta}) = [p_{jk}(\boldsymbol{\theta})]$ be the predicted probability matrix. Then the loss function is defined as:

$$\mathcal{L}(\boldsymbol{\theta}) = (1 - \lambda)\mathcal{L}(\mathbb{P}^r(\boldsymbol{\theta})) + \lambda\,\mathcal{L}(\mathbb{P}^s(\boldsymbol{\theta})),$$

where $\mathcal{L}(\mathbb{P}^r(\boldsymbol{\theta}))$ and $\mathcal{L}(\mathbb{P}^s(\boldsymbol{\theta}))$ are the loss functions associated with relevance and serendipity, respectively, and λ is a weighting factor that balances the two objectives.

However, training the model directly on both relevance and serendipity data is often challenging, as serendipity labels are typically scarce and incomplete. To address this issue, we adopt a transfer learning strategy inspired by (Pandey et al., 2018b). The two-stages ID-based model training process is illustrated in Figure 2. We begin by training the ID-based sequence model using a large-scale dataset with only relevance labels. During this stage, we optimize the relevance loss:

$$\mathcal{L}(\mathbb{P}^r(\boldsymbol{\theta})) = -\sum_{j,k} \Big(r_{jk} \log p^r_{jk}(\boldsymbol{\theta}) \\ + (1 - r_{jk}) \log(1 - p^r_{jk}(\boldsymbol{\theta})) \Big),$$

where $p^r_{jk}(\boldsymbol{\theta})$ here is the predicted relevance probability for the user–item pair (u_j, i_k).

Once the model is well-trained on the relevance task. We perform fine-tuning on a smaller dataset that includes serendipity annotations[1]. During this phase, all parameters except the last dense layer are frozen to avoid overfitting on the scarce serendipity labeled data, and we minimize the serendipity loss function:

$$\mathcal{L}(\mathbb{P}^s(\boldsymbol{\theta})) = -\sum_{j,k} \Big(s_{jk} \log p^s_{jk}(\boldsymbol{\theta}) \\ + (1 - s_{jk}) \log\big(1 - p^s_{jk}(\boldsymbol{\theta})\big) \Big),$$

where $p^s_{jk}(\boldsymbol{\theta})$ here is the predicted serendipity probability for the user–item pair (u_j, i_k).

---

[1] Human annotation data on serendipity is scarce within the community. We have collected all publicly available labels to date.

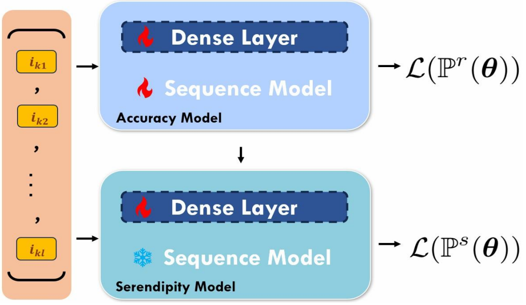


Figure 2: ID-based Model Training: Our training process consists of two stages: initial training on a large-scale relevance-labeled dataset, and subsequent fine-tuning on a smaller, serendipity-labeled dataset.

3.2 Fine-tuning Data Generation

A key innovation in our approach is the generation of fine-tuning data specifically designed for instruct-tuning the LLM. In our reranking approach, we utilize an instruction-following paradigm. The users' historical interaction data H is formatted into natural language patterns embedded with template T. This template serves as a natural language instruction for the LLMs, guiding them to understand the context and produce the desired reranking results. The following sections detail this process, including Serendipity Prompting and Refinement, and Listwise Ranking, all of which contribute to refining the output.

3.2.1 Serendipity Prompting and Refinement

To enable the LLM to understand and evaluate serendipity in recommendations, we employ a two-step prompting approach. First, we introduce a set of specially designed prompts to build an initial understanding of serendipity, guiding the model to identify recommendations that are both aligned with user interests and unexpectedly pleasant. Second, we refine this understanding through in-context learning with a quantitative scoring system (ranging from 1 to 5). We provide the LLM with examples of user interactions, including both positive (high scores) and negative (low scores) examples, to represent user preferences and evaluate serendipity.

Now, we aim to select an item $i \in I$ that balances between relevance $\mathbb{P}(i \mid H)$ and serendipity. Let $S(i)$ represent the serendipity score of item i. Although we do not introduce an explicit parameter to control the trade-off, the prompt design guides the LLM to consider both factors simultaneously. Thus, the chosen item is given by:

$$\hat{i} = \arg\max_{i \in I} \big(\mathbb{P}(i \mid H) \cdot S(i)\big).$$

The scoring system further enhances this balance by providing concrete metrics that help the LLM identify items that are both relevant and serendipitous.

3.2.2 Listwise Ranking: Parsing the Output of LLMs

After establishing the serendipity-aware prompting scheme, we extend the concept to listwise ranking. Our approach begins by retrieving a set of candidates $C = \{c_1, c_2, \ldots, c_m\}$ through a candidate generation model. Here m represents the number of candidate items (e.g., $m = 20$).

Unlike traditional methods with a predefined candidate order, our approach generates multiple ranking orders to balance serendipity and accuracy by guiding the LLM to explore diverse possibilities. We aim to reorder C as $C' = \{C_{l_1}, C_{l_2}, \ldots, C_{l_k}\}$, where $k \leq m$, to maximize both user relevance and serendipity:

$$\mathcal{L} = \arg\max_{C' \subseteq C} \mathbb{E}\left[U(C' \mid H)\right],$$

where $U(C' \mid H)$ represents the utility function implicitly learned by the LLM from the user's historical interactions H and serendipity-oriented prompts.

To ensure consistency and reliability, the LLM also summarizes user preferences and provides brief explanations during the ranking process.

Through this process, each step—reformatting user histories, introducing serendipity prompts, applying scoring systems, and exploring multiple ranking arrangements—yields concrete examples that feed back into the training of the LLM. In this way, the data we generate from these interactions is not merely demonstration data, but a carefully crafted resource for instruct-tuning the LLM, ensuring that it learns to naturally and consistently incorporate serendipity into its recommendations.

3.3 Instruction Dataset Construction and LLM Fine-tuning

To enable large language models (LLMs) to suggest serendipitous recommendations with human perception alignment, we need to translate these tasks into text-based instructions.

3.3.1 Explainable Generative Recommendation and the SUN Framework

In our task, we focus on a core objective: Explainable Generative Recommendation. To achieve this, we propose a new fine-tuning framework called **SUN** (Recommendation Specialized Unified Tuning Network). The **SUN** framework is designed to enable LLMs to understand users' diverse needs and generate corresponding recommendations with text-based instructions. In this process, we introduce a triplet $(\mathbf{H}, \mathbf{E}, \mathbf{T})$ to define the key components of the recommendation task, including User History $\mathbf{H}$, Engagement Profile $\mathbf{E}$, and Task Format $\mathbf{T}$:

$$\mathbf{SUN} = (\mathbf{H}, \mathbf{E}, \mathbf{T})$$

The definition of each component is defined as follows: **History (H)** represents users' historical information, such as browsed items, or viewed content. This data reveal patterns of interest and long-term preferences.

Engagement Profile (E) is a key element used for modeling and interpreting user behavior. It is derived by first inferring user preferences and intentions using the fine-tuned ID-based sequential recommendation model with LLM reranker. Then, a powerful LLM is employed to interpret these inferences, generating nuanced explanations that capture the user's engagement with the recommended items. This process ensures it to capture both **preferences (P)** (e.g., product categories, brands, styles) and **intentions (I)** (e.g., purchase intent behind a search), helping the model provide more accurate and personalized recommendations. The engagement profile (E) can be further subdivided based on preference information (e.g., no preference, general preference, novelty preference) and the clarity and specificity of user intentions. Detailed definitions of the engagement profile and the prompts used to drive the LLM for profile generation are provided in **Appendix C** and Tables 21, 22 and 23.

Task (T) denotes the specific tasks to be performed by LLM. We define four task types:

Generative Recommendation: The model directly generates recommendations, rather than selecting from existing candidates. This allows the model to create novel recommendations based on the user's history and engagement profile.

Direct Recommendation: The model selects the most suitable item from a predefined set of candidates. This approach quickly chooses the best match without generating new candidates.

Reranking: Given a set of candidate items, the model reorders them according to a specific objective, such as optimizing for accuracy or serendipity.

Matching: The model evaluates whether a given candidate item matches the user's preferences or intentions, producing a binary answer "Yes" or "No".

To further enhance the explainability and generalization of the recommendations, we introduce $\mathbf{SUN}^{-1}$, a reverse task mechanism represented by a similar triplet:

$$\mathbf{SUN}^{-1} = (\mathbf{H}, \mathbf{E}, \mathbf{T}^{-1})$$

where H and E are the same as above, while $\mathbf{T}^{-1}$ denotes the reverse task. While $\mathbf{SUN}$ utilizes the history (H) and engagement profile (E) to perform the four recommendation tasks (Generative Recommendation, Direct Recommendation, Reranking, and Matching), $\mathbf{SUN}^{-1}$ operates in reverse. It takes the history (H) and the results of the four recommendation tasks as input and infers the user's engagement profile (E). This reverse process helps to refine the understanding of user needs, purpose and preference by analyzing how the model's recommendations align with the user's engagement.

3.3.2 Template-Based Construction and Unified Refinement of Engagement Profile

Within the $\mathbf{SUN}$ framework, we need to define a set of general templates for common recommendation scenarios, such as item selection, suggestion generation, and candidate reranking. These templates are then populated with previously generated user engagement data.

By contrast, the $\mathbf{SUN}^{-1}$ framework focuses on explaining user preferences and behaviors. To achieve this, we develop specialized templates that guide the model in finding the reason behind the recommendations and identifying patterns in user engagement.

Leveraging these two types of templates, the model integrates stable preferences and dynamic intentions into a unified profile, capturing both the user's consistent behaviors and evolving goals. As new data becomes available, this engagement profile is continuously refined. The refinement incorporates explicit feedback (e.g., ratings, reviews) and recent activities, allowing the model to adapt and ensure that personalized recommendations remain both accurate and contextually relevant over time.

3.4 Similarity-based Mapping

To map LLM's generated items to our actual item set, we use Levenshtein distance as a simple text similarity metric for computational efficiency. When the LLM produces a candidate $\hat{i}$, we compute the Levenshtein distance between it and all items in I.

4 Experiments

In this section, we conduct experiments to answer the following research questions:

- **RQ1:** How does **SOLAR** perform compared with current LLM-based recommendation models?
- **RQ2:** How effective is our hybrid fine-tune and rerank strategy compared with traditional ID-based recommendation models?
- **RQ3:** How do the different components in **SOLAR** affect its effectiveness?

Due to space constraints, complete experimental results are provided in Appendix.

4.1 Experimental Setup

4.1.1 Datasets

We conduct experiments on three real-world datasets from different domains: MovieLens, Amazon Books, and Amazon Movies&TV. These datasets provide distinct item titles and text-based reviews, enabling LLMs to leverage both. Note that after an extensive search for large-scale relevance-labeled datasets with real user-labeled serendipity data, we found that **these datasets are currently the only publicly available ones with user-labeled serendipity data**. The detailed stats of whole instruction set structured by these datasets using our method are presented in Table 1.

1. **MovieLens:** A classic movie recommendation dataset (Harper and Konstan, 2015) with user ratings, movie descriptions, and serendipity labels

Instruct type	Movielens	Books	Movies & TV
User History ($\mathbf{H}$)	19040	16185	16824
Engagement Profile ($\mathbf{E}$)	114240	97110	100944
Rec Task ($\mathbf{T}$)	57120	48555	50472
ReverseRec Task ($\mathbf{T}^{-1}$)	76160	64740	67296
ALL Tasks ($\mathbf{T}+\mathbf{T}^{-1}$)	133280	113295	117768

Table 1: Detailed Stats Of Datasets

for a subset of reviews from the Serendipity 2018 dataset (Kotkov et al., 2018).

2. **Books:** A large-scale dataset from *Amazon Review* (Ni et al., 2019) containing book metadata, user reviews, ratings, and serendipity labels from the SerenLens dataset (Fu et al., 2023).

3. **Movies&TV:** A dataset from *Amazon Review* with movie/TV metadata, user reviews, ratings, and serendipity labels from the SerenLens dataset.

4.1.2 Evaluation Metrics

To evaluate generative recommendation, we adopt a leave-one-out evaluation strategy, following previous work on sequential recommendation. We adopt the widely used metric **HR** and **NDCG** for accuracy measurement and $\mathbf{HR}_{seren}$ and $\mathbf{NDCG}_{seren}$ for serendipity measurement.

1. **HR@K:** Measures the fraction of cases that the ground-truth next item appears among the top K recommendations, over the given m candidates, where $K \leq m$.

2. **NDCG@K:** Evaluates ranking quality by assigning higher importance to correct items that appear among the top K recommendations.

3. $\mathbf{HR}_{seren}$**@K:** A serendipity-based version of **HR**. It checks whether an item is relevant and whether it provides a surprising or novel experience to the user among top K recommendations.

4. $\mathbf{NDCG}_{seren}$**@K:** A serendipity-based version of **NDCG**.

The formulation of $\mathbf{HR}_{seren}$ and $\mathbf{NDCG}_{seren}$ are defined as:

$$HR_{\text{seren}}@k = \frac{1}{|U|}\sum_{u\in U}\mathbb{I}\left(\sum_{i=1}^{k} S_i > 0\right)$$

where $|U|$ represents the total number of users. S_i is the serendipity score of the item at position i, typically 0 or 1, indicating whether the item provides serendipity. $\mathbb{I}$ is an indicator function that returns 1 if $\sum_{i=1}^{k} S_i > 0$ (i.e., there is at least one serendipitous item in the top k), and 0 otherwise.

$$NDCG_{\text{seren}}@k = \sum_{i=1}^{k}\frac{S_i}{\log_2(i+1)}$$

where S_i represents the serendipity score of the item at position i, typically valued as 0 or 1.

4.1.3 Baselines

We compare **SOLAR** against state-of-the-art ID and LLM-based recommendation baselines:

1. **SASRec:** (Kang and McAuley, 2018) A causal sequential model using a unidirectional transformer to predict the next item in a sequence of IDs.

2. **BERT4Rec:** (Sun et al., 2019) A sequential method using a bidirectional transformer to learn user behavior sequences for recommendations.

3. **TALLRec:** (Bao et al., 2023) A framework that fine-tunes LLMs for recommendation tasks, aligning pre-trained models with recommendation.

4. **P5:** (Geng et al., 2022) A training framework for T5, extended to LLMs. It uses personalized prompting and template-based training to unify multiple recommendation tasks.

5. **LLMRank:** (Hou et al., 2024) An LLM-based recommendation system that pairs with sequential models with careful prompting.

6. **RecLM:** (Lu et al., 2024) A framework combining supervised and reinforcement learning to improve LLMs' instruction-following abilities and generalize across various recommendation tasks.

7. **LlamaRec:** (Yue et al., 2023b) A two-stage recommendation framework based on LLMs. It first retrieves candidate items using an efficient sequential recommender and then ranks them using an LLM.

8. **GPT-4o:** (Achiam et al., 2023) A state-of-the-art general purpose large language model from OpenAI.

4.2 Performance Comparison: SOLAR vs. Baselines (RQ1)

Performance comparison of SOLAR against both traditional ID-based and LLM-based baselines is summarized in Table 2.

Traditional ID-based models (e.g., SASRec, BERT4Rec) effectively capture collaborative signals, leading to strong accuracy. However, they struggle with serendipitous recommendations due to the inherent scarcity of serendipitous interactions and the complexity of modeling unexpected yet relevant recommendations.

LLM-based methods (e.g., P5, TALLRec, LLMRank, RecLM, GPT-4o) leverage language understanding but lack domain-specific adaptation, limiting their ability to model serendipity effectively. For instance, TALLRec's binary classification approach focuses only on relevance, preventing it from capturing nuanced user preferences for surprising recommendations. P5, relying on simple multi-task learning, fails to account for complex serendipitous interactions. LLMRank and GPT-4o, using a general-purpose LLM without fine-tuning, lacks personalization due to its limited domain knowledge, while RecLM, despite achieving reasonable accuracy, struggles with serendipity due to the scarcity of labeled data in this aspect.

In contrast, SOLAR directly addresses these limitations through its design. By integrating a serendipity-finetuned ID model with an LLM reranker for serendipity-labeled data augmentation, SOLAR mitigates the data imbalance issue that constrains models like RecLM. Additionally, its SUN framework introduces a diverse and complex set of recommendation tasks, enabling a deeper understanding of user preferences compared to the simpler multi-task approaches of P5. Furthermore, SOLAR's domain-adaptive instruction tuning refines the LLM for recommendation-specific tasks, overcoming the lack of domain alignment in LLMRank and GPT-4o. As a result, SOLAR achieves strong accuracy while significantly improving serendipity performance.

Model	**Movielens**		**Movies&TV**		**Books**	
	Acc	**Seren**	**Acc**	**Seren**	**Acc**	**Seren**
SASRec	0.1936	0.0568	0.1478	0.0181	**0.1383**	0.0194
BERT4Rec	0.1616	0.0341	0.1369	0.0084	0.1304	0.0146
P5	0.0234	0.0138	0.0398	0.0118	0.0323	0.0089
TALLRec	0.0310	0.0141	0.0341	0.0103	0.0385	0.0149
LLMRank	0.0603	0.0219	0.0584	0.1207	0.0537	0.0187
RecLM	0.1353	0.0894	**0.1591**	0.1131	0.1012	0.0719
Llamarec	0.1712	0.0483	0.1127	0.0492	0.0995	0.0333
GPT-4o	0.1853	0.0395	0.1256	0.0443	0.1097	0.0496
SOLAR	**0.2160**	**0.1284**	0.1451	**0.1314**	0.1203	**0.0902**

Table 2: Comparison of **SOLAR** against baselines on the MovieLens, Movies & TV, and Books datasets

4.3 Analysis of Sequential Model Reranking Effectiveness (RQ2)

We evaluate our hybrid approach (serendipity-tuned SASRec with LLM reranking, serving as the intermediate model in the construction of SOLAR) against four baselines: **SASRec**, **BERT4Rec**, and their serendipity-tuned variants. Evaluations use standard metrics (**HR@K**, **NDCG@K**) and their serendipity counterparts, with K = 1, 5, 10. As shown in Table 3, our method achieves superior performance in both accuracy and serendipity metrics on MovieLens and Movies&TV datasets, while obtaining the best serendipity scores on Books. Figure 3 illustrates that unlike baselines which trade accuracy for serendipity, our method maintains competitive accuracy while improving serendipity across all datasets. Detailed results are presented in Tables 7 and 8.

Model	**Movielens**		**Movies&TV**		**Books**	
	Acc	**Seren**	**Acc**	**Seren**	**Acc**	**Seren**
SASRec	0.0298	0.0000	**0.0494**	0.0081	0.0518	0.0000
SASRec-seren	0.0259	0.0114	0.0371	0.0068	0.0293	0.0049
BERT4Rec	0.0281	0.0000	0.0190	0.0000	**0.0348**	0.0000
BERT4Rec-seren	0.0273	0.0114	0.0086	0.0000	0.0106	0.0049
Our reranker	**0.0303**	**0.0114**	0.0455	**0.0098**	0.0298	**0.0049**

Table 3: Comparative of our hybrid reranker approach against baseline models (**SASRec**, **BERT4Rec**, and their serendipity-tuned variants).

Model	**Movielens**		**Movies&TV**		**Books**	
	Acc	**Seren**	**Acc**	**Seren**	**Acc**	**Seren**
SOLAR	**0.2160**	**0.1284**	**0.1451**	**0.1314**	**0.1203**	**0.0902**
w/o reranker	0.1524	0.1101	0.0915	0.0949	0.0962	0.0598
w/o SM	0.1828	0.1193	0.1072	0.1168	0.0909	0.0824
w/o SM & reranker	0.1386	0.0780	0.0878	0.0584	0.1077	0.0255
w/o SUN	0.0363	0.0398	0.0413	0.0487	0.0527	0.0312
NoAugment	0.1354	0.0734	0.0928	0.0401	0.1123	0.0333

Table 4: Comparison of **SOLAR** against ablated variants.

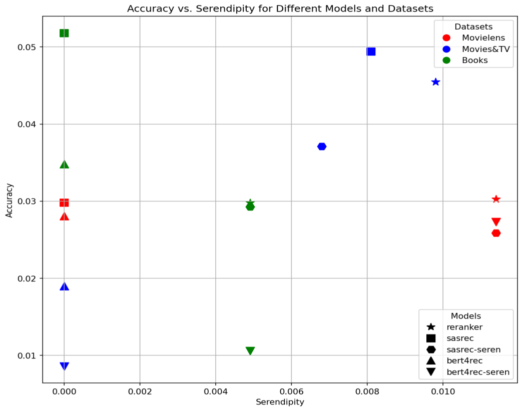


Figure 3: Trade-off between accuracy (**HR/NDCG@1**) and serendipity (**HRseren/NDCGseren@1**) across recommendation models and datasets. The star symbol denotes our proposed method.

4.4 Ablation Study (RQ3)

To assess the contribution of each component in **SOLAR**, we conduct an ablation study with the following variants:

1. **SOLAR**: The full model with all components.

2. **SOLAR**$_{w/o\ reranker}$: Removes the LLM reranker, relying only on the serendipity-tuned **SASRec**$_{seren}$ for data generation.
3. **SOLAR**$_{w/o\ SM}$: Retains the LLM reranker but replaces the serendipity-tuned model with a standard sequential model optimized for accuracy.
4. **SOLAR**$_{w/o\ SM\ \&\ reranker}$: Removes both the reranker and the serendipity model, reducing the framework to a standard accuracy-based recommendation model.
5. **SOLAR**$_{w/o\ SUN}$: Eliminates the **SUN** framework, retaining only the recommendation task.
6. **NoAugment**: Conduct experiments using only user-labeled unaugmented data.

Table 4 shows that **SOLAR** outperforms all ablated variants across both accuracy and serendipity metrics.

Our ablation study demonstrates that **each component of SOLAR plays a crucial role in balancing accuracy and serendipity**. The LLM reranker significantly enhances recommendation diversity and novelty, while the serendipity-tuned model contributes to both serendipity and overall recommendation quality. Removing either component leads to noticeable performance degradation, with the impact being more pronounced when both are removed. The SUN framework proves to be the most critical element, as its removal results in the largest decline in both accuracy and serendipity, highlighting its importance in integrating textual and collaborative signals. **Overall, the full SOLAR model achieves the best trade-off between accuracy and serendipity, underscoring the necessity of incorporating all components for effective recommendations.**

4.5 Proportion Control

To explore how the ratio of augmented data impacts both accuracy and serendipity. We utilized 364,343 SFT data entries for fine-tuning. Experiments were conducted with four augmentation ratios: 0 (using the raw, un-fine-tuned LLaMA model), 0.33, 0.66, and 1.0, representing 0%, 33%, 66%, and 100% of augmented data relative to the original training set, respectively.

The experimental results are as follows:

Model	**Movielens**		**Movies&TV**		**Books**	
	Acc	**Seren**	**Acc**	**Seren**	**Acc**	**Seren**
UnfinetunedLlama (0)	0.1284	0.0675	0.0978	0.0403	0.1214	0.0317
SOLAR (0.33)	0.1651	0.0953	0.1154	0.0949	0.1147	0.0538
SOLAR (0.66)	0.1944	0.1156	0.1349	0.1166	0.1184	0.0748
SOLAR	**0.2160**	**0.1284**	**0.1451**	**0.1314**	**0.1203**	**0.0902**

Table 5: Proportion control of **SOLAR** on different data augmentation ratios.

Results show a general trend of improvement in both Acc and Serendipity as the augmentation ratio increases, particularly pronounced in Serendipity. However, the improvement rate diminishes as the ratio approaches 1.0, suggesting a potential performance plateau.

4.6 A/B Test

Due to resource constraints, we conducted an offline A/B test based on user surveys to evaluate the proposed model's performance, particularly in enhancing recommendation serendipity. Participants were presented with recommendations from three methods: A1 (baseline), A2 (random), and B (our model), and rated the "relevance" and "serendipity" of each on a 5-point Likert scale, without knowing the source. To mitigate priming effects, relevance was rated before serendipity.

Results showed that our model (Group B) significantly outperformed both A1 and A2 in terms of both serendipity and relevance ($p < 0.05$), with a significant positive correlation observed between the two within Group B. For detailed setup, data analysis, and results, refer to the Appendix.

4.7 Case Study

Table 6 in Appendix presents a real-world case study illustrating how recommendations can be generated based on a user's historical viewing history. In this scenario, the user's previously watched films (e.g., The Witch , Sea Inside) reflect a preference for works characterized by rich storytelling, emotional depth, and socially relevant themes. Given a set of candidate movies (e.g., 7 Women , Without Bias) including target item (e.g., Remember), each model (**GPT4o**, **RecLM**, and **SOLAR**) was tasked with selecting five recommended items from candidate items. By comparing the resulting recommendation lists. Table 6 highlights key differences in recommendation approaches. GPT4o over-explains, burying recommendations in excessive detail. RecLM hallucinates, suggesting movies outside the candidate set. In contrast, SOLAR effectively and accurately identifying the target item with a concise recommendation list that surpasses the other two models.

5 Conclusion

In summary, this work introduces **SOLAR**, a serendipity-optimized LLM-based recommendation framework that addresses the domain gap and label scarcity challenges in producing diverse and surprising recommendations. By integrating an serendipity fine-tuned ID-based model, LLM-based reranking, and domain-adaptive instruction tuning (**SUN**), **SOLAR** significantly enhances both accuracy and serendipity, as confirmed by experiments on multiple real-world datasets. The results demonstrate **SOLAR**'s capability to break free from filter bubbles, offering more user-centric, engaging, and genuinely novel recommendations.

Limitations

Although **SOLAR** demonstrates improved accuracy and serendipity, it still faces several potential risks and limitations. A potential risk lies in popularity bias, where the model may favor frequently appearing items in training data, though this could be mitigated through diversity-aware sampling strategies. Additionally, while our similarity-based generative recommendation method effectively maps LLM outputs to existing items, it may constrain the system's ability to suggest truly novel items. Furthermore, using LLMs for large-scale recommendation tasks may face efficiency challenges. In this work, we evaluate on the reranking task, where the input size is limited, and thus, noticeable efficiency issues do not arise. However, this constraint may limit its applicability in computationally restricted settings. We conduct experiments to evaluate efficiency, and the results can be found in the Appendix H. Future work could explore more efficient data collection methods, incorporate broader feedback signals, and develop techniques that support open-ended item generation without relying on predefined catalogs.

Ethics Statement

Our **SOLAR** framework aims to enhance recommendation diversity while maintaining user privacy and fairness. We rely on anonymized historical data and adhere to data protection standards. While serendipity may influence user preferences, we will strive to avoid biases and harmful content. Ongoing monitoring, transparency about recommendation processes, and allowing users to adjust or opt out of personalized suggestions help ensure ethical and responsible deployment.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. TALLRec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, RecSys '23, page 1007–1014. ACM.

Minmin Chen, Yuyan Wang, Can Xu, Ya Le, Mohit Sharma, Lee Richardson, Su-Lin Wu, and Ed Chi. 2021. Values of user exploration in recommender systems. In *Proceedings of the 15th ACM Conference on Recommender Systems*, pages 85–95.

Daixuan Cheng, Yuxian Gu, Shaohan Huang, Junyu Bi, Minlie Huang, and Furu Wei. 2024. Instruction pre-training: Language models are supervised multitask learners. *arXiv preprint arXiv:2406.14491*.

Jorge Díez, David Martínez-Rego, Amparo Alonso-Betanzos, Oscar Luaces, and Antonio Bahamonde. 2019. Optimizing novelty and diversity in recommendations. *Progress in Artificial Intelligence*, 8:101–109.

Michael D Ekstrand and Martijn C Willemsen. 2016. Behaviorism is not enough: better recommendations through listening to users. In *Proceedings of the 10th ACM conference on recommender systems*, pages 221–224.

Yue Feng, Shuchang Liu, Zhenghai Xue, Qingpeng Cai, Lantao Hu, Peng Jiang, Kun Gai, and Fei Sun. 2023. A large language model enhanced conversational recommender system. *arXiv preprint arXiv:2308.06212*.

Zhe Fu, Xi Niu, and Li Yu. 2023. Wisdom of crowds and fine-grained learning for serendipity recommendations. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 739–748.

Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. 2023. Chat-rec: Towards interactive and explainable llms-augmented recommender system. *arXiv preprint arXiv:2303.14524*.

Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. 2010. Beyond accuracy: evaluating recommender systems by coverage and serendipity. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 257–260.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as language processing (RLP): A unified pretrain, personalized prompt & predict paradigm (P5). In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 299–315.

Shijie Geng, Juntao Tan, Shuchang Liu, Zuohui Fu, and Yongfeng Zhang. 2023. VIP5: Towards multimodal foundation models for recommendation. *arXiv preprint arXiv:2305.14302*.

F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19.

Yupeng Hou, Zhankui He, Julian McAuley, and Wayne Xin Zhao. 2023. Learning vector-quantized item representation for transferable sequential recommenders. In *Proceedings of the ACM Web Conference 2023*, pages 1162–1171.

Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2024. Large language models are zero-shot rankers for recommender systems. In *European Conference on Information Retrieval*, pages 364–381. Springer.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

Denis Kotkov, Joseph A Konstan, Qian Zhao, and Jari Veijalainen. 2018. Investigating serendipity in recommender systems based on real user feedback. In *Proceedings of the 33rd annual acm symposium on applied computing*, pages 1341–1350.

Denis Kotkov, Shuaiqiang Wang, and Jari Veijalainen. 2016. A survey of serendipity in recommender systems. *Knowledge-Based Systems*, 111:180–192.

Chen Li, Yixiao Ge, Jiayong Mao, Dian Li, and Ying Shan. 2023a. Taggpt: Large language models are zero-shot multimodal taggers. *arXiv preprint arXiv:2304.03022*.

Jinming Li, Wentao Zhang, Tian Wang, Guanglei Xiong, Alan Lu, and Gerard Medioni. 2023b. GPT4rec: A generative framework for personalized recommendation and user interests interpretation. *arXiv preprint arXiv:2304.03879*.

Pan Li and Alexander Tuzhilin. 2020. Latent unexpected recommendations. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(6):1–25.

Xiangyang Li, Bo Chen, Lu Hou, and Ruiming Tang. 2023c. Ctrl: Connect collaborative and language model for ctr prediction. *arXiv preprint arXiv:2306.02841*.

Yaoyiran Li, Xiang Zhai, Moustafa Alzantot, Keyi Yu, Ivan Vulić, Anna Korhonen, and Mohamed Hammad. 2024. Calrec: Contrastive alignment of generative llms for sequential recommendation. *arXiv preprint arXiv:2405.02429*.

Xinyu Lin, Wenjie Wang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. 2023. A multi-facet paradigm to bridge large language model and recommendation. *arXiv preprint arXiv:2310.06491*.

Junling Liu, Chao Liu, Peilin Zhou, Renjie Lv, Kang Zhou, and Yan Zhang. 2023. Is ChatGPT a good recommender? a preliminary study. *arXiv preprint arXiv:2304.10149*.

Wensheng Lu, Jianxun Lian, Wei Zhang, Guanghua Li, Mingyang Zhou, Hao Liao, and Xing Xie. 2024. Aligning large language models for controllable recommendations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8159–8172, Bangkok, Thailand. Association for Computational Linguistics.

Yucong Luo, Mingyue Cheng, Hao Zhang, Junyu Lu, and Enhong Chen. 2023. Unlocking the potential of large language models for explainable recommendations. *arXiv preprint arXiv:2312.15661*.

Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. 2023. Zero-shot listwise document reranking with a large language model. *arXiv preprint arXiv:2305.02156*.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 188–197.

Lin Ning, Luyang Liu, Jiaxing Wu, Neo Wu, Devora Berlowitz, Sushant Prakash, Bradley Green, Shawn O'Banion, and Jun Xie. 2024. User-llm: Efficient llm contextualization with user embeddings. *arXiv preprint arXiv:2402.13598*.

Gaurav Pandey, Denis Kotkov, and Alexander Semenov. 2018a. Recommending serendipitous items using transfer learning. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 1771–1774.

Gaurav Pandey, Denis Kotkov, and Alexander Semenov. 2018b. Recommending serendipitous items using transfer learning. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 1771–1774, New York, NY, USA. Association for Computing Machinery.

Zachary A Pardos and Weijie Jiang. 2020. Designing for serendipity in a university course recommendation system. In *Proceedings of the tenth international conference on learning analytics & knowledge*, pages 350–359.

Eli Pariser. 2011. *The filter bubble: What the Internet is hiding from you*. penguin UK.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. Rankvicuna: Zero-shot listwise document reranking with open-source large language models. *arXiv preprint arXiv:2309.15088*.

Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Representation learning with large language models for recommendation. In *Proceedings of the ACM on Web Conference 2024*, pages 3464–3475.

Mario Rodriguez, Christian Posse, and Ethan Zhang. 2012. Multiple objective optimization in recommender systems. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 11–18.

Lior Rokach and Slava Kisilevich. 2012. Initial profile generation in recommender systems using pairwise comparison. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):1854–1859.

Anna Sepliarskaia, Julia Kiseleva, Filip Radlinski, and Maarten de Rijke. 2018. Preference elicitation as an optimization problem. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 172–180.

Leheng Sheng, An Zhang, Yi Zhang, Yuxin Chen, Xiang Wang, and Tat-Seng Chua. 2024. Language models encode collaborative signals in recommendation. *arXiv preprint arXiv:2407.05441*.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Bohao Wang, Feng Liu, Jiawei Chen, Yudi Wu, Xingyu Lou, Jun Wang, Yan Feng, Chun Chen, and Can Wang. 2024a. Llm4dsr: Leveraing large language model for denoising sequential recommendation. *arXiv preprint arXiv:2408.08208*.

Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, et al. 2023a. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity. *arXiv preprint arXiv:2310.07521*.

Hangyu Wang, Jianghao Lin, Xiangyang Li, Bo Chen, Chenxu Zhu, Ruiming Tang, Weinan Zhang, and Yong Yu. 2023b. Alt: Towards fine-grained alignment between language and ctr models for click-through rate prediction. *arXiv preprint arXiv:2310.19453*.

Jianling Wang, Haokai Lu, Yifan Liu, He Ma, Yueqi Wang, Yang Gu, Shuzhou Zhang, Shuchao Bi, Lexi Baugher, Ed Chi, et al. 2024b. Llms for user interest exploration: A hybrid approach. *arXiv preprint arXiv:2405.16363*.

Lei Wang, Songheng Zhang, Yun Wang, Ee-Peng Lim, and Yong Wang. 2023c. Llm4vis: Explainable visualization recommendation using chatgpt. *arXiv preprint arXiv:2310.07652*.

Yidan Wang, Zhaochun Ren, Weiwei Sun, Jiyuan Yang, Zhixiang Liang, Xin Chen, Ruobing Xie, Su Yan, Xu Zhang, Pengjie Ren, et al. 2024c. Enhanced generative recommendation via content and collaboration integration. *arXiv preprint arXiv:2403.18480*.

Yunjia Xi, Weiwen Liu, Jianghao Lin, Xiaoling Cai, Hong Zhu, Jieming Zhu, Bo Chen, Ruiming Tang, Weinan Zhang, Rui Zhang, et al. 2023. Towards open-world recommendation with knowledge augmentation from large language models. *arXiv preprint arXiv:2306.10933*.

Chengxing Xie, Canyu Chen, Feiran Jia, Ziyu Ye, Kai Shu, Adel Bibi, Ziniu Hu, Philip Torr, Bernard Ghanem, and Guohao Li. 2024. Can large language model agents simulate human trust behaviors? *arXiv preprint arXiv:2402.04559*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.

Li Yang, Anushya Subbiah, Hardik Patel, Judith Yue Li, Yanwei Song, Reza Mirghaderi, and Vikram Aggarwal. 2024. Item-language model for conversational recommendation. *arXiv preprint arXiv:2406.02844*.

Zhenrui Yue, Sara Rabhi, Gabriel de Souza Pereira Moreira, Dong Wang, and Even Oldridge. 2023a. Llamarec: Two-stage recommendation using large language models for ranking. *arXiv preprint arXiv:2311.02089*.

Zhenrui Yue, Sara Rabhi, Gabriel de Souza Pereira Moreira, Dong Wang, and Even Oldridge. 2023b. Llamarec: Two-stage recommendation using large language models for ranking. *arXiv preprint arXiv:2311.02089*.

An Zhang, Yuxin Chen, Leheng Sheng, Xiang Wang, and Tat-Seng Chua. 2024a. On generative agents in recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1807–1817.

Junjie Zhang, Yupeng Hou, Ruobing Xie, Wenqi Sun, Julian McAuley, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2024b. Agentcf: Collaborative learning with autonomous language agents for recommender systems. In *Proceedings of the ACM on Web Conference 2024*, pages 3679–3689.

Yang Zhang, Fuli Feng, Jizhi Zhang, Keqin Bao, Qifan Wang, and Xiangnan He. 2023. Collm: Integrating collaborative embeddings into large language models for recommendation. *arXiv preprint arXiv:2310.19488*.

Hongke Zhao, Songming Zheng, Likang Wu, Bowen Yu, and Jing Wang. 2024. Lane: Logic alignment of non-tuning large language models and online recommendation systems for explainable reason generation. *arXiv preprint arXiv:2407.02833*.

Zihuai Zhao, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, et al. 2023. Recommender systems in the era of large language models (llms). *arXiv preprint arXiv:2307.02046*.

Appendix A: Case Study

A real-world case study illustrating how recommendations can be generated based on a user's historical viewing history in different models.

Appendix B: Implementation Details

Inspired by recent advances in automatic prompting strategies (Cheng et al., 2024; Xu et al., 2023), we employ GPT-4(Achiam et al., 2023) to generate instruction data by constructing user engagement profiles. We use LoRA (Hu et al., 2021) to fine-tune the LLaMA 3.1 8B model (Touvron et al., 2023) with Adam Optimizer (Kingma and Ba, 2017) on the generated triplet (H, E, T) in the **SUN** framework. The learning rate is set to 1×10^{-3}, the LoRA dimension is set to 8, LoRA alpha to 16, and the batch size is 32. We conducted the training and finetuning process on 2 Nvidia A100 GPUs using PyTorch framework.

To effectively evaluate LLMs' ability to capture user preferences, we conduct our evaluation in a reranking setting rather than direct generation. This is because when LLMs generate recommendations as free-form text, the potential candidate space consists of all possible text sequences, which can be vastly larger than the actual set of available items. For fair comparison, all models are evaluated on their ability to rank 20 explicitly provided candidates to select 5 items for recommendation. Specifically, for **TALLRec**, we compute **NDCG** and **HR** by processing batches of m candidate items, where the model produces logits to rank candidates from highest ("Yes") to lowest ("No"). Similarly, for **P5** and **LLMrank** , we input the same 20 candidates and evaluate their ranking outputs using **NDCG** and **HR**.

We employ two ID-based sequential recommendation models **SASRec** and **BERT4Rec**, training them on the three datasets (MovieLens, Books, and Movies & TV) to compute user-item interaction accuracy. All model parameters are tuned for each dataset with the optimal dataset hyperparameters.

After this initial training phase, we fine-tune each model on serendipity-labeled data from the same datasets to improve their ability to generate serendipitous recommendations. In this stage, only the final feed-forward layer is fine-tuned to incorporate the serendipity objective, while the rest of the network remains fixed.

Appendix C: Detailed Results

Table 7 and Table 8 presents the detailed results of our proposed fine-tune and rerank approach against traditional ID-based baselines.

Table 9 and Table 10 presents the detailed results of **SOLAR** against LLM-based baselines.

Table 11 and Table 12 provides detailed results of **SOLAR** against all ablated variants.

Input Example	
Instruction	You are an expert recommender engine. You need to select a recommendation list considering user's historical interactions. The historical interactions are provided as follows: The Witch , Sea Inside , Long Way Round , Project Nim , Lady Bird , Best of Youth , La meglio gioventù , Take Me Home , The Hunting Ground . The candidate items are: Idle Class, The , 7 Women , Great Santini, The , Love(Szerelem) , Shadow of the Vampire , We Are the Best! (Vi är bäst! , April 9th , Remember , Without Bias , Elvira, Mistress of the Dark . Please select a recommendation list with 5 different items from candidate items.
GPT4o	To generate a recommendation list, we can consider the user's historical interactions to identify patterns in their preferences. Based on the provided historical data, the user appears to enjoy movies that are critically acclaimed, have emotional depth, explore human experiences, or involve impactful storytelling. Using these preferences, here's a curated list of recommendations: Love(Szerelem) , Shadow of the Vampire , Idle Class, The , Remember , Without Bias .
RecLM	7 Women , Without Bias , Remember , Elvira, Mistress of the Dark , Hurricane of Fun
SOLAR	Remember , Love (Szerelem) , Great Santini, The , We Are the Best! (Vi är bäst! , 7 Women

Table 6: Comparison of recommendation lists generated by **GPT4o**, **RecLM**, and our **SOLAR** method. Each model selects items based on the user's historical interactions and candidate pool. Compared to **GPT4o** and **RecLM**, **SOLAR** provides a more aligned and diverse set of recommendations, effectively balancing accuracy and serendipity to yield the best overall results. We highlight texts for history items , candidate items , target item and unpredicted item .

Dataset	**Metric**	**SASRec**	**BERT4Rec**	**SASRec-Seren**	**BERT4Rec-Seren**	**Ours**
MovieLens	HR@1	0.0298	0.0281	0.0259	0.0273	**0.0303**
	HR@5	**0.0974**	0.0997	0.0831	0.0924	0.0806
	HR@10	0.1518	**0.1598**	0.1359	0.1497	0.1410
	NDCG@1	0.0298	0.0224	0.0224	0.0271	**0.0303**
	NDCG@5	**0.0635**	0.0525	0.0525	0.0593	0.0493
	NDCG@10	0.0810	0.0695	0.0695	0.0777	**0.0831**
Books	HR@1	**0.0518**	0.0403	0.0293	0.0106	0.0498
	HR@5	0.1057	0.0903	0.0661	0.0823	**0.1174**
	HR@10	0.1069	0.1295	0.0834	0.1149	**0.1184**
	NDCG@1	0.0518	0.0356	0.0317	0.0316	**0.0498**
	NDCG@5	**0.0805**	0.0635	0.0495	0.0570	0.0609
	NDCG@10	**0.0906**	0.0762	0.0551	0.0675	0.0884
Movies & TV	HR@1	0.0427	0.0190	0.0371	0.0086	**0.0455**
	HR@5	**0.0764**	0.0563	0.0752	0.0106	0.0603
	HR@10	0.0953	0.0824	0.0928	0.0276	**0.1127**
	NDCG@1	0.0427	0.0226	0.0446	0.0106	**0.0455**
	NDCG@5	0.0779	0.0395	0.0612	0.0192	**0.0798**
	NDCG@10	0.0844	0.0479	0.0669	0.0240	**0.0892**

Table 7: Comparison of our hybrid approach against different models in terms of accuracy.

Appendix D: Detailed Implementation and Results of A/B Test

Participants. We recruited 53 random participants for this study.

Procedure. Participants were presented with user profiles (including browsing history) and corresponding recommendations generated by each of the three methods: A1, A2, and B. To avoid bias, the source of the recommendation (A1, A2, or B) was not revealed to the participants. Each participant was shown the user profile and recommendations, and asked to rate the recommendations based on two metrics: relevance and serendipity. The concept of "relevance" and "serendipity" were clearly explained to the participants before the test. We employed a 5-point Likert scale for collecting the ratings. To minimize potential priming effects where explicitly considering relevance might influence

Dataset	Metric	SASRec	BERT4Rec	SASRec-Seren	BERT4Rec-Seren	Ours
MovieLens	HR@1	0.0000	0.0000	0.0114	0.0114	**0.0114**
	HR@5	0.0341	0.0227	0.0455	0.0341	**0.0227**
	HR@10	0.0455	0.0227	**0.0568**	0.0455	0.0455
	NDCG@1	0.0000	0.0000	0.0114	0.0114	**0.0114**
	NDCG@5	0.0160	0.0227	0.0263	0.0234	**0.0163**
	NDCG@10	0.0195	0.0227	0.0301	0.0268	**0.0234**
Books	HR@1	0.0000	0.0000	0.0049	0.0049	**0.0049**
	HR@5	0.0049	0.0000	0.00194	0.0194	**0.0194**
	HR@10	0.0146	0.0097	0.0340	0.0243	**0.0340**
	NDCG@1	0.000	0.0000	0.0049	0.0049	**0.0049**
	NDCG@5	0.0031	0.0000	0.0115	**0.0116**	0.0115
	NDCG@10	0.0060	0.0033	0.0167	0.0133	**0.0167**
Movies & TV	HR@1	0.0081	0.000	0.0068	0.0000	**0.0098**
	HR@5	0.0081	0.0081	**0.0170**	0.0081	0.0146
	HR@10	0.0242	0.0081	0.0292	0.0081	**0.0146**
	NDCG@1	0.0081	0.0000	0.0068	0.0000	**0.0098**
	NDCG@5	0.0081	0.0035	0.0119	0.0051	**0.0106**
	NDCG@10	0.0131	0.0035	**0.0157**	0.0051	0.0146

Table 8: Comparison of our hybrid approach against different models in terms of serendipity.

Dataset	Metric	P5	TALLRec	LLMRank	RecLM	SOLAR
MovieLens	HR@1	0.0234	0.0310	0.0603	0.1353	**0.2160**
	HR@5	0.1138	0.1483	0.2885	0.3517	**0.3787**
	NDCG@1	0.0234	0.0310	0.0603	0.1353	**0.2160**
	NDCG@5	0.0975	0.0922	0.1513	0.2405	**0.2950**
Movies & TV	HR@1	0.0398	0.0341	0.0584	**0.1591**	0.1451
	HR@5	0.1074	0.1453	0.2670	**0.4001**	0.3574
	NDCG@1	0.0398	0.0341	0.0584	**0.1591**	0.1451
	NDCG@5	0.1549	0.1395	0.2261	**0.2785**	0.2501
Books	HR@1	0.0323	0.0385	0.0537	0.1012	**0.1203**
	HR@5	0.0974	0.1003	0.1377	0.3076	**0.3259**
	NDCG@1	0.0323	0.0385	0.0537	0.1012	**0.1203**
	NDCG@5	0.0915	0.1043	0.1594	0.2037	**0.2214**

Table 9: Comparison of SOLAR against baselines in terms of accuracy.

the perception of serendipity, the relevance question was presented before the serendipity question. Participants were not shown the serendipity question until after completing the relevance assessment.

Groups. The control group (A) included two subgroups: recommendations from a baseline algorithm (A1) and random recommendations (A2). The experimental group (B) received recommendations generated by our LLM-based model.

- **A1**: Baseline recommendation algorithm [Provide a brief description of the algorithm].
- **A2**: Random recommendations [Explain how the random recommendations were generated].
- **B**: Our proposed LLM-based model [Provide a brief description, but no need to go into extensive detail since it's covered in the main text].

Evaluation Metrics.

Relevance (Positive). Participants rated the relevance of each recommendation by answering: "Based on the user's browsing history, how relevant is this recommendation to their interests?" (1: Not at all relevant, 5: Highly relevant).

Serendipity (Positive). Participants rated their agreement with the statement: "This recommendation is surprising and delightful" (1: Strongly Disagree, 5: Strongly Agree).

Data Analysis. We employed the Mann-Whitney U test to compare the serendipity and relevance scores between groups (A1 vs. B and A2 vs. B). We used

Dataset	Metric	P5	TALLRec	LLMRank	RecLM	SOLAR
MovieLens	HR@1	0.0138	0.0141	0.0219	0.0894	**0.1284**
	HR@5	0.0512	0.0514	0.1544	0.2546	**0.2959**
	NDCG@1	0.0138	0.0141	0.0219	0.0894	**0.1284**
	NDCG@5	0.0635	0.0673	0.0525	0.1676	**0.2115**
Movies & TV	HR@1	0.0118	0.0103	0.1207	0.1131	**0.1314**
	HR@5	0.0492	0.0636	0.1963	0.3467	**0.3613**
	NDCG@1	0.0118	0.0103	0.1207	0.1131	**0.1314**
	NDCG@5	0.0524	0.0653	0.1370	0.2311	**0.2397**
Books	HR@1	0.0089	0.0149	0.0187	0.0716	**0.0902**
	HR@5	0.0567	0.0585	0.1547	0.2735	**0.3039**
	NDCG@1	0.0089	0.0149	0.0187	0.0716	**0.0902**
	NDCG@5	0.0564	0.0581	0.1088	0.1696	**0.1990**

Table 10: Comparison of SOLAR against baselines in terms of serendipity.

Dataset	Metric	w/o reranker	w/o SM	w/o SM&reranker	w/o SUN	SOLAR
MovieLens	HR@1	0.1524	0.1828	0.1386	0.0363	**0.2160**
	HR@5	0.3085	0.3363	0.2874	0.2536	**0.3787**
	NDCG@1	0.1524	0.1828	0.1386	0.0363	**0.2160**
	NDCG@5	0.2284	0.2574	0.2098	0.1463	**0.2950**
Movies & TV	HR@1	0.0915	0.1072	0.0878	0.0413	**0.1451**
	HR@5	0.3017	0.3147	0.2951	0.2387	**0.3574**
	NDCG@1	0.0915	0.1072	0.0878	0.0413	**0.1451**
	NDCG@5	0.1945	0.2103	0.1892	0.1869	**0.2501**
Books	HR@1	0.0962	0.0909	0.1077	0.0527	**0.1203**
	HR@5	0.3155	0.3050	0.3255	0.2431	**0.3259**
	NDCG@1	0.0962	0.0909	0.1077	0.0527	**0.1203**
	NDCG@5	0.2037	0.1958	0.2153	0.1656	**0.2214**

Table 11: Comparison of SOLAR against ablated variants in terms of accuracy.

Dataset	Metric	w/o reranker	w/o SM	w/o SM&reranker	w/o SUN	SOLAR
MovieLens	HR@1	0.1101	0.1193	0.0780	0.0398	**0.1284**
	HR@5	0.2867	0.2615	0.2982	0.1972	**0.2959**
	NDCG@1	0.1101	0.1193	0.0780	0.0398	**0.1284**
	NDCG@5	0.1961	0.1854	0.1851	0.1386	**0.2115**
Movies & TV	HR@1	0.0949	0.1168	0.0584	0.0487	**0.1314**
	HR@5	0.2883	0.3321	0.2719	0.2279	**0.3613**
	NDCG@1	0.0949	0.1168	0.0584	0.0487	**0.1314**
	NDCG@5	0.1915	0.2199	0.1625	0.1525	**0.2397**
Books	HR@1	0.0598	0.0824	0.0255	0.0312	**0.0902**
	HR@5	0.2951	0.3176	0.2667	0.1794	**0.3039**
	NDCG@1	0.0598	0.0824	0.0255	0.0312	**0.0902**
	NDCG@5	0.1761	0.1990	0.1436	0.1092	**0.1990**

Table 12: Comparison of SOLAR against ablated variants in terms of serendipity.

Spearman's rank correlation coefficient to assess the relationship between serendipity and relevance scores within each group (A1, A2, and B). A significance level of $p < 0.05$ was used.

Experimental Results.

Rating Data. Table 13 presents the raw rating data collected from the participants.

Mann-Whitney U Test and Spearman Correla-

Table 13: Raw Serendipity and Relevance Ratings

Group	Serendipity_1	Serendipity_2	Serendipity_3	Serendipity_4	Serendipity_5	Relevance_1	Relevance_2	Relevance_3	Relevance_4	Relevance_5
A1	2	16	13	14	8	3	8	15	23	4
A2	6	14	19	8	6	9	16	17	7	4
B	4	3	12	15	19	1	7	7	17	21

tion Coefficients Results. Table 14 presents the results of the Mann-Whitney U test and the Spearman correlation coefficients, respectively.

Discussion. The results demonstrate that our method (Group B) achieved significantly higher serendipity scores compared to both the baseline algorithm (A1) and random recommendations (A2) (Mann-Whitney U test, $p < 0.01$). Group B also scored significantly higher on relevance compared to both A1 and A2 (Mann-Whitney U test, $p < 0.05$). Importantly, within Group B, we observed a statistically significant positive correlation between serendipity and relevance scores (Spearman's rho = 0.28, $p < 0.05$). This suggests that the LLM is capable of generating recommendations that are both surprising and relevant to users' interests, even when relevance is assessed prior to serendipity.

Limitations. The offline nature of this A/B test has inherent limitations. The sample size of 53 participants, while sufficient for initial validation, may not fully represent the broader user population. Additionally, subjective ratings may be influenced by individual biases. Future work should involve a larger-scale online A/B test to further validate these findings in a real-world setting.

Appendix E: Detailed Categorization of Preferences and Intentions

User engagement profiling serves as a crucial step in understanding personalized recommendation scenarios. By examining a user's historical interactions alongside their expressed or inferred preferences, we can more accurately capture their long-term interests and current intentions. In this appendix, we provide a detailed categorization of user preferences and intentions, expanding on the definitions presented in the main text. This additional information aims to clarify how these categories can be applied to construct more nuanced user engagement profiles, ultimately leading to more effective and explainable recommendation outcomes.

Preference (P) describes a user's personalized likes or dislikes for certain product attributes or features. Preferences capture inherent, long-term interests and needs. Depending on the level of personalization, user preferences can be categorized as follows:

- *No Preference (P0)*: When the system lacks any information about the user's preferences, recommendations are non-personalized. This often occurs in cold-start situations where the system has no historical data to base recommendations on.

- *General Preferences (PC)*: Reflect interests through both direct expressions and inferred patterns expressed by the user. This includes straightforward preferences expressed through ratings and reviews, providing direct feedback. It also includes patterns observed from long-term interactions, such as browsing history and purchase activities, which reveal underlying interests. Together, these aspects form a comprehensive view of the user's personalized likes and dislikes.

- *Novelty Preferences (PN)*: Reflect the user's interest in exploring both new and unexpected content beyond their typical preferences. This includes a willingness to actively try categories or domains different from their usual choices. It also reflects an openness to items that pleasantly surprise them, even if these items do not match their established tastes. These elements together add diversity and exploration to recommendations.

Intention (I) describes a user's immediate needs and goals at a specific point in time. Unlike long-term preferences, intentions focus on the user's current, specific demands, which may differ from their usual interests. Intentions can be categorized based on their level of clarity:

- *No Intention (I0)*: The user has no clear needs, showing exploratory behavior to discover potential interests through the system's recommendations.

- *General Intention (IC)*: Reflect the user's expressed need, which can range from vague to specific. This intention can be vague, where the user describes a general goal or purpose without identifying specific product types, attributes, or features. Such expressions often lack clear guidance, requiring further refinement or exploration. Alternatively, the intention can be specific, where the user provides detailed information, explicitly outlining the characteristics, attributes, or requirements they are seeking.

- *Exploratory Intention (IE)*:Reflect the user's desire to explore and engage with new domains or product types. This intention demonstrates a purposeful approach where the user actively searches for opportunities to broaden their knowledge, experience diverse options, or discover innovative solutions that expand their understanding and satisfaction. It highlights a proactive and goal-oriented behavior in their exploration process.

Appendix F: Construction of SUN and SUN^{-1}

In this appendix, the two figures (Figure 4 and Figure 5), we illustrate two examples of the overall process framework that transforms user interaction records and system instructions into recommendation outputs. In the first part (yellow), the system will receive instructions

Table 14: Mann-Whitney U Test and Spearman Correlation Coefficients Results

Table 15: Mann-Whitney U Test

Comparison	U Value	p-value
A1 vs. B - Serendipity	980.0	0.002913257250473625
A2 vs. B - Serendipity	809.0	5.581316015540164e-05
A1 vs. B - Relevance	920.5	0.000729550390629055
A2 vs. B - Relevance	606.5	1.2171162949477135e-07

Table 16: Spearman Correlation

Group	rho	p-value
A1	0.043	0.760
A2	-0.178	0.203
B	0.283	0.040

Metric	**Value (seconds)**
Average Latency	2.13
P50 Latency	2.68
P90 Latency	2.74
P95 Latency	2.77
P99 Latency	2.98
QPS (Queries per Second)	10.48

Table 17: Latency performance metrics using vLLM on two A100 GPUs.

as input. In the second part (red), the user engagement profile serves as the foundation, combining the user's history with candidate items and dynamically selecting the most relevant next recommendation through various methods at output (green), including generative, direct recommendation, reranking, and matching. As the reverse method, the user history and potential recommendation results serve as input (red) to prompt the model to output the user engagement profile (green). And detailed templates are presented in Table 18, Table 19 and Table 20.

Appendix G: Templates of Generation of Engagement Profile

In this appendix, we present detailed templates for generating engagement profiles based on multiple datasets, including Movielens, Booksand Movies & TV, see Table 21,Table 22, and Table 23. These templates leverage users' historical interaction data (and corresponding reviews) to extract and infer various dimensions of user preferences and intentions. In these templates, {interaction} and {reviews} serve as placeholders for user interaction histories and associated feedback, while {constraint} introduces necessary limiting conditions.Using these templates, the system improves recommendation accuracy and serendipity.

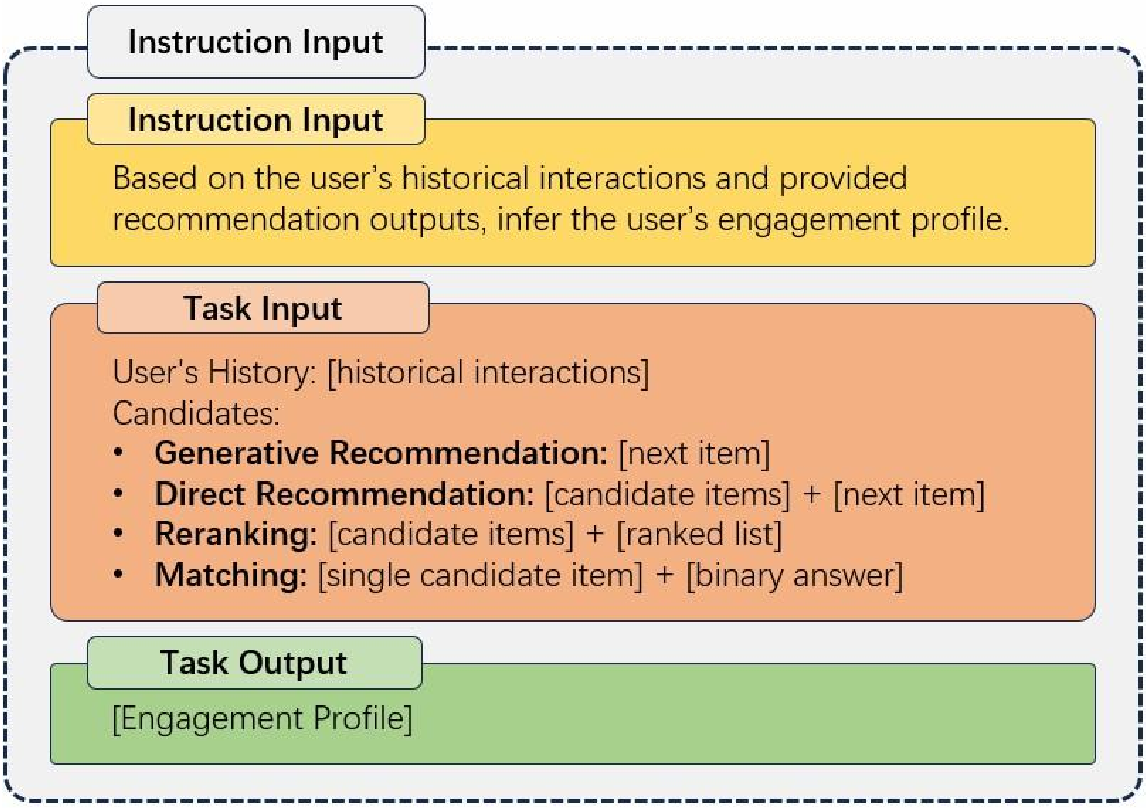
Appendix H: Latency Evaluation and Scalability Considerations

To assess the scalability of our approach, we conducted latency experiments using **two NVIDIA A100 GPUs** with **vLLM deployment** under simple setup configurations. Results are summarized in Table 17.

While these results indicate limitations for real-time applications in an academic setup, we believe performance can be significantly improved through industrial-grade infrastructure and optimization. Potential strategies include:

- **Infrastructure Scaling**: Deploying across more GPUs with proper load balancing.
- **Model Optimization**: Quantization, distillation, or using smaller LLM variants.
- **Caching Strategies**: Implementing result caching for frequent queries.
- **Batch Processing**: Leveraging more aggressive batch inference for improved throughput.

We acknowledge scalability challenges and discuss them in the limitation section. However, successful industrial deployments of LLM-based recommender systems demonstrate that, with proper infrastructure and optimization, these issues can be effectively addressed.



Based on the user's historical interactions and provided recommendation outputs, infer the user's engagement profile.

User's History: [historical interactions]
Candidates:

- **Generative Recommendation:** [next item]
- **Direct Recommendation:** [candidate items] + [next item]
- **Reranking:** [candidate items] + [ranked list]
- **Matching:** [single candidate item] + [binary answer]

[Engagement Profile]

Figure 4: Reverse Recommendation Task.

	Templates of Construction of SUN (RecTempalte)
(1)	`{reranking}` The behavioral sequence of the user is shown below: `{historical_interactions}`, which can be used to infer the user's preferences `{explicit_preference}`. Then please rerank the items to better align with the user's preferences by comparing the candidates and their similarities to the user's preferences. The candidates are: `{candidate_items}`.
(2)	`{reranking}` You have some information about this user, which is shown below: `{explicit_preference}`, the user's historical interactions: `{historical_interactions}` Based on this information, please recommend the reranking order of items for the user, which should match the user's preference, from the following candidates: `{candidate_items}`
(3)	`{generation}` Using the user's historical interactions as input data, predict the next product that the user is most likely to interact with. The historical interactions are provided as follows: `{historical_interactions}`.
(4)	`{generation}` Given the user's interaction history: `{historical_interactions}`, what is the optimal product to suggest next?
(5)	`{generation}` Given the sequence of the user's past interactions: `{historical_interactions}`, what is the most suitable product to recommend next?
(6)	`{generation}` Considering the user's interaction pattern: `{historical_interactions}`, suggest the next likely product they would engage with.
(7)	`{generation}` Given the historical context of user interactions: `{historical_interactions}`, what is the optimal next product recommendation?
(8)	`{generation}` Based on the user's historical engagement data: `{historical_interactions}`, provide the next product recommendation.
(9)	`{generation}` Based on the user's past interaction data: `{historical_interactions}`, suggest the most relevant product for their next interaction.
(10)	`{generation}` Using the provided interaction history: `{historical_interactions}`, determine the most likely product the user would engage with next.
(11)	`{generation}` You are a recommendation system, and are good at recommending products to a user based on his preferences. Given the user's preferences: `{explicit_preference}`, please recommend products that are consistent with those preferences.
(12)	`{generation}` As we know, a user's behavior is driven by his preferences, which determine what they are likely to buy next. Your task is to predict what products a user will purchase next, based on his preferences. Given the user's preferences as follows: `{explicit_preference}`, please make your prediction.
(13)	`{generation}` Given the following historical interaction of the user: `{historical_interactions}`. You can infer the user's preference: `{explicit_preference}`. Please predict next possible item for the user.
(14)	`{generation}` To make a recommendation for this user, we need to analyze their historical interactions: `{historical_interactions}`. As we know, historical interactions reflect the user's preferences `{explicit_preference}`. Based on these preferences, please recommend an item that you think would be suitable for them.
(15)	`{generation}` Recommend the next potential product to a user based on his profile and past interactions. You have access to the user's profile information, including his preference: `{explicit_preference}` and past interactions: `{historical_interactions}`. For example, if the user recently interacted with `{recent_item}`, you might consider similar products. Now, based on this approach, determine what product would be recommended to him next.
(16)	`{generation}` Imagine the user recently interacted with `{recent_item}`. Using this example, and given the user's historical interactions as input data: `{historical_interactions}`, predict the next product that the user is most likely to interact with.
(17)	`{direct}` The user has previously purchased the following items: `{historical_interactions}`. This information indicates their personalized preferences `{explicit_preference}`. Based on this information, is it likely that the user will interact with `{candidate_item}` next?
(18)	`{direct}` Based on the user's historical interaction list, which is provided as follows: `{historical_interactions}`, you can infer the user's personalized preference `{explicit_preference}`. And your task is to use this information to predict whether the user will click on `{candidate_item}` next.
(19)	`{direct}` Please recommend an item to the user based on the following information about the user: `{historical_interactions}`, the user's historical interaction, which is as follows: `{explicit_preference}` Try to select one item from the following candidates that is consistent with the user's preference: `{candidate_items}`.
(20)	`{generation}` Suppose you are a search engine, now the user search that `{explicit_preference_vague_intention_specific_intention}`, can you generate the item to respond to user's query?

Table 18: Generation templates for the Recommendation Task

	Templates of Construction of SUN (RecTemplateInverse)
(1)	`{explicit_preference}` The behavioral sequence of the user is shown below: `{historical_interactions}`. The candidates were provided as: `{candidate_items}`, and they have been reranked to better align with the user's preferences: `{rerank_list}`. Based on this information, please infer the user's explicit preferences that likely led to this reranking.
(2)	`{explicit_preference}` You have observed that the user has clicked on the following items: `{historical_interactions}`. The following candidates were presented: `{candidate_items}`, and they have been reranked in an order deemed suitable for the user: `{rerank_list}`. Based on this information, please infer the user's explicit preferences that likely led to this reranking.
(3)	`{explicit_preference}` You have some information about this user, which is shown below: the user's historical interactions: `{historical_interactions}`. The candidates presented were: `{candidate_items}`, and they have been reranked in the following order: `{rerank_list}`. Based on this information, please infer the user's explicit preferences that would justify this reranking.
(4)	`{implicit_preference}` The user has interacted with the following items in the past: `{historical_interactions}`. The candidates provided were: `{candidate_items}`, and they have been reranked to better align with the user's interests: `{rerank_list}`. Based on this information, please infer the user's implicit preferences that likely led to this reranking.
(5)	`{vague_intention}` The user has shown the following historical interactions: `{historical_interactions}`, and the candidate items were provided as: `{candidate_items}`. The candidates have been reranked in this order: `{rerank_list}`. Based on this information, infer the user's vague intention that could explain why this reranking aligns with their preferences.
(6)	`{specific_intention}` Analyzing the user's past behavior: `{historical_interactions}` and the given candidates: `{candidate_items}`, which have been reordered to: `{rerank_list}`, please determine the user's specific intention that could explain this preference for certain elements over others.
(7)	`{explicit_preference}` Given the following historical interaction of the user: `{historical_interactions}`. And the next recommended item: `{next_item}`. Please infer the user's explicit preferences that would likely lead to this recommendation.
(8)	`{novelty_preference}` Given the user's historical behavior and intention: `{historical_interactions}`, and the next recommended item: `{next_item}`, please infer the user's exploratory preferences that would justify this recommendation.
(9)	`{specific_intention}` Given the following historical interactions of the user: `{historical_interactions}`, and the next recommended item: `{next_item}`. Please infer the specific intention that would likely lead to this recommendation, such as seeking a particular genre, theme, or type of item.
(10)	`{specific_intention}` To better understand the user's needs, consider their past interactions: `{historical_interactions}`. The next recommended item is: `{next_item}`. Based on this information, infer the user's specific intention that would justify this recommendation, focusing on concrete preferences or desires.
(11)	`{exploratory_intention}` The user has recently been recommended the following item: `{next_item}`. Given the user's historical actions: `{historical_interactions}` and the candidates: `{candidate_items}`, please infer the user's exploratory intention that would justify this surprising recommendation.
(12)	`{exploratory_intention}` The user was recommended the following item: `{next_item}`. Considering their historical interactions: `{historical_interactions}` and the set of candidates: `{candidate_items}`, please infer the user's lack of specific intention for surprising recommendations that justify the selection of this item.
(13)	`{explicit_preference}` Please try to infer the preference to the user based on the following information: `{historical_interactions}`, the user's historical interaction, which is as follows: `{next_item}` and the candidate item: `{candidate_items}`.
(14)	`{vague_intention}` The user has received the following recommendation: `{next_item}`. Given their historical actions: `{historical_interactions}` and the set of candidates: `{candidate_items}`, please infer the user's vague intention that could justify this recommendation.
(15)	`{implicit_preference}` Based on the user's historical interaction list: `{historical_interactions}`, and considering the candidate items: `{candidate_items}`, the item most likely to be clicked next is: `{next_item}`. Please infer the user's implicit preferences that would justify the selection of this item.

Table 19: Generation templates for the Reverse Recommendation Task

	Templates of Construction of SUN (RecTemplateSeren)
(1)	`{generation}` The user likes to explore new types of products and has recently shown interest in items that differ from their usual preferences. The user is looking to try new domains or product types. Based on the user's historical behavior and intention: `{historical_interactions}`, generate a product recommendation that aligns with the user's novelty preference: `{novelty_preference}`.
(2)	`{generation}` The user is interested in exploring new types of products while maintaining certain explicit preferences: `{explicit_preference}`. Given the user's exploratory intention (`{exploratory_intention}`) to try something new and different, please generate a product recommendation that aligns with both the user's explicit preferences and their desire for exploration.
(3)	`{direct}` The user enjoys receiving surprising recommendations and wants to try items that do not match their usual preferences. Based on the user's exploratory intention:`{exploratory_intention}` and combine the user's historical action : `{historical_interactions}`, select the item most likely to offer a pleasant surprise from the following candidates: `{candidate_items}`
(4)	`{matching}` The user is interested in new types of products that do not match their usual preferences:`{explicit_preference}` but their needs are still unclear. Please determine whether the following item matches the user's vague exploratory intention and answer "Yes" or "No": `{candidate_item}`
(5)	`{direct}` The user has no specific intention but enjoys receiving surprising recommendations. Based on this, select the item most likely to provide a pleasant surprise from the following candidates: `{candidate_items}`
(6)	`{matching}` The user enjoys being surprised and has shown implicit preferences based on their historical interactions: `{historical_interactions}`. The user's current intention may be vague as following : `{vague_intention}`. Based on this information, evaluate the following candidate item: `{candidate_item}` to determine if it would be a suitable recommendation for the user, please answer "Yes" or "No" for the fitness of candidate.
(7)	`{generation}` You are a search engine. Here is the historical interaction of a user: `{historical_interactions}`. And his personalized preferences are as follows: `{explicit_preference}`. Your task is to generate a new product that are consistent with the user's preference.
(8)	`{generation}` The user has interacted with a list of items, which are as follows: `{historical_interactions}`. Based on these interacted items, the user current intent are as follows `{vague_intention}`, and your task is to generate products that match the user's current intent.
(9)	`{generation}` As a search engine, you are assisting a user who is searching for the query: `{specific_intention}`. Your task is to recommend products that match the user's query and also align with their preferences based on their historical interactions, which are reflected in the following: `{historical_interactions}`
(10)	`{direct}` Using the user's current query: `{explicit_preference_vague_intention_specific_intention}` and their historical interactions: `{historical_interactions}` you can estimate the user's preferences `{explicit_preference}`. Please respond to the user's query by selecting an item from the following candidates that best matches their preference and query: `{candidate_items}`
(11)	`{direct}` The user wants to try some products and searches for: `{explicit_preference_vague_intention_specific_intention}`. In addition, they have previously bought: `{historical_interactions}`. You can estimate their preference by analyzing his historical interactions. `{explicit_preference}` Please recommend one of the candidate items below that best matches their search query and preferences: `{candidate_items}`

Table 20: Generation templates for the RecTemplate for Serendipity Purpose

	Templates of Generation of Engagement Profile (Movielens)
(1)	User's historical interactions: `{interaction}`. Based on these movie titles, use your knowledge to generate a description of the user's implicit preferences, such as their favorite genres, themes, or notable patterns. `{constraint}`
(2)	The user has browsed the following movies in chronological order: `{interaction}`. Based on this browsing history, use your understanding of these movies to generate a description of the user's implicit preferences, including their likely favorite genres, themes, or types of movies. `{constraint}`
(3)	Recently, the user has browsed the following movies: `{interaction}`. Based on this recent activity, apply your knowledge of these movie to generate a description of the user's current movie preferences, focusing on genres, themes, or other noticeable patterns.`{constraint}`
(4)	Analyze the user's recent viewing history: `{interaction}`. From these interactions, use your knowledge of these movies to infer the user's implicit preferences, such as preferred genres, sub-genres, or specific types of storylines. `{constraint}`
(5)	The user has shown a strong interest in the following movies: `{interaction}`. Using this data, infer their explicit preferences, such as particular themes, moods, or types of narratives they actively seek.`{constraint}`
(6)	Consider the user's engagement with the following movies: `{interaction}`. Based on these patterns, determine their explicit preferences, such as favorite directors, frequent actors, or recurring motifs that they seem to appreciate. `{constraint}`
(7)	The user has recently browsed a variety of different movie genres: `{interaction}`. Based on this diverse viewing pattern, describe the user's novelty preferences, such as their openness to exploring new genres or trying unexpected movie types. `{constraint}`
(8)	Given the user's browsing history: `{interaction}`, identify any novelty preferences they may have, such as a willingness to explore genres outside their usual interest or a desire for unique and unconventional film experiences. `{constraint}`
(9)	The user has moved from browsing typical genres to less common ones: `{interaction}`. Describe the user's novelty preferences, focusing on their interest in discovering diverse genres or unique cinematic styles. `{constraint}`
(10)	The user has recently watched the following movies: `{interaction}`. Reflect on this history to infer a general type or mood of movies they might be interested in next, without narrowing down to a specific genre or characteristic. `{constraint}`
(11)	Given the user's recent viewing history: `{interaction}`, suggest a broad intention for what they may want to watch next, focusing on an overall style or feeling rather than pinpointing a particular movie or specific genre. `{constraint}`
(12)	Based on these movies: `{interaction}`, generate an open-ended intention that represents a general mood or broad category the user could be leaning towards, even if their specific preferences aren't clear. `{constraint}`
(13)	The user has watched these movies: `{interaction}`. Use this data to determine a specific movie intention they might have, such as seeking a particular genre, a specific plot, or a film with certain defining characteristics. `{constraint}`
(14)	Based on the user's recent movie list: `{interaction}`, infer a clearly defined intention for the next type of film they may want to watch, focusing on particular elements like genre, theme, or distinctive features. `{constraint}`
(15)	Considering the user's viewing pattern: `{interaction}`, determine a specific intention about the next movie they are likely to watch, including precise details about the genre, mood, or main elements they are interested in. `{constraint}`
(16)	The user has recently watched these movies:`{interaction}`. Based on this history, suggest an exploratory intention where the user might want to explore genres or types of movies they haven't typically watched. `{constraint}`
(17)	Given the user's movie-watching history of: `{interaction}`, infer an exploratory intention indicating their curiosity to explore new and different genres, styles, or narrative types that they might not have considered before. `{constraint}`
(18)	Using the following interaction data: `{interaction}`, generate an exploratory intention for the user, where they express interest in trying out new genres, themes, or movie types that differ from their usual choices. `{constraint}`

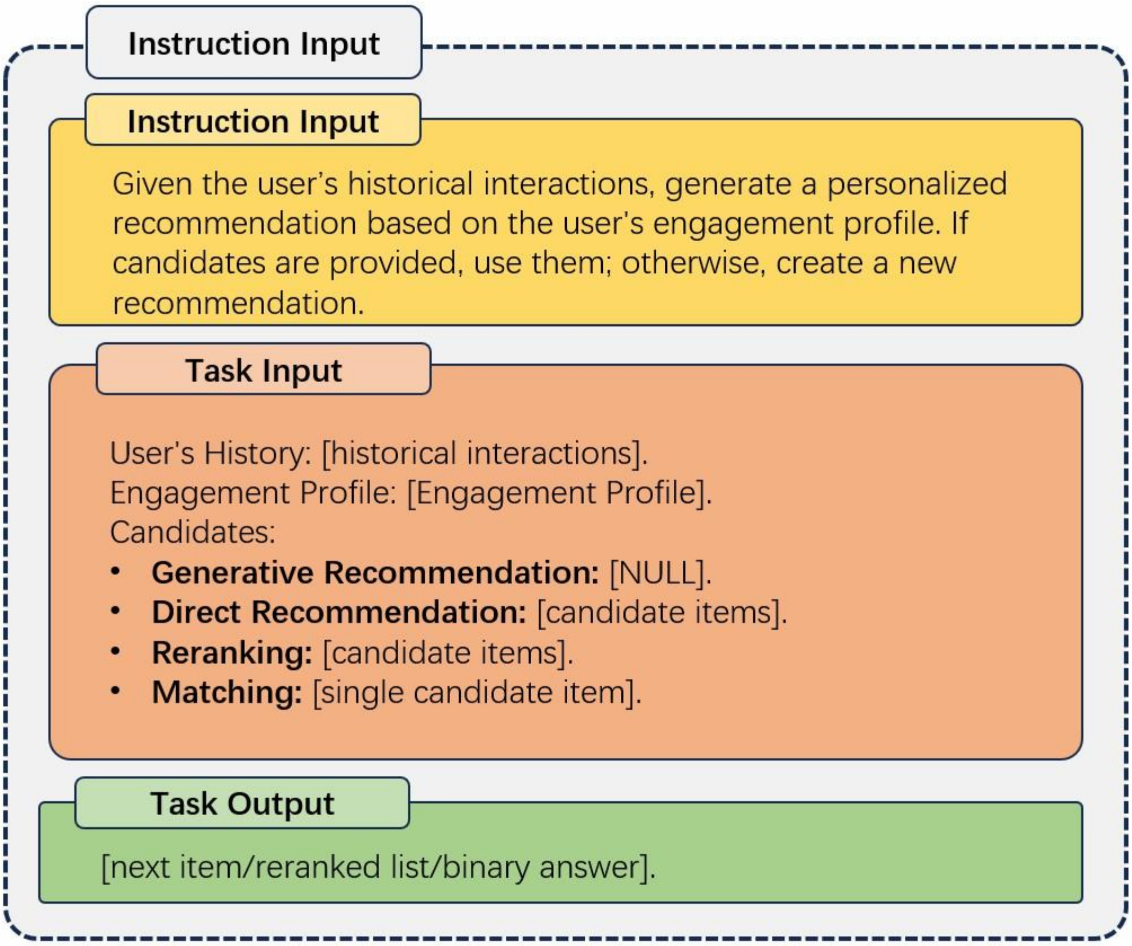
Table 21: Generation templates for the Movielens dataset

	Templates of Generation of Engagement Profile (Books)
(1)	Analyze the user's reading history: `{interaction}` and the associated reviews: `{reviews}`. From these data points, determine the user's explicit preferences, such as the genres, themes, or specific book characteristics they explicitly praise or mention in their comments. `{constraint}`
(2)	Considering the user's reading history of these books: `{interaction}`, along with their corresponding reviews: `{reviews}`, generate a description of the user's explicit preferences, focusing on any recurring genres, themes, or patterns evident in their comments. `{constraint}`
(3)	Based on the user's recent engagement with the following books: `{interaction}` and their comments: `{reviews}`, identify their explicit preferences by analyzing the sentiments and focus of their reviews, such as preferred genres, themes, or author styles they frequently mention or praise. `{constraint}`
(4)	Analyze the user's reading history: `{interaction}` and the associated reviews: `{reviews}`. From these data points, determine the user's explicit preferences, such as the genres, themes, or specific book characteristics they explicitly praise or mention in their comments. `{constraint}`
(5)	The user has shown a clear interest in certain books: `{interaction}`, with specific comments: `{reviews}`. Using this data, infer their explicit preferences, such as favorite themes, plot types, or narrative styles they often highlight in their reviews. `{constraint}`
(6)	Consider the user's engagement with these books: `{interaction}`, accompanied by their reviews: `{reviews}`. Based on these reviews, identify explicit preferences, such as preferred authors, frequent genres, or writing styles that the user frequently praises or critiques. `{constraint}`
(7)	The user has recently reviewed a variety of different genres or unconventional books: `{interaction}`, with comments: `{reviews}`. Describe the user's novelty preferences, such as their openness to experimenting with new genres or exploring unique literary styles, based on the diversity of their reviews. `{constraint}`
(8)	Given the user's diverse reading history: `{interaction}` and their reviews: `{reviews}`, identify any novelty preferences they may have, such as a tendency to seek out unique literary experiences or genres that are outside their usual interests. `{constraint}`
(9)	The user has moved from reading typical genres to exploring less common ones: `{interaction}`, as indicated by their reviews: `{reviews}`. Describe the user's novelty preferences, focusing on their interest in discovering new genres or unconventional narrative approaches. `{constraint}`
(10)	The user has recently read the following books: `{interaction}`, with the following reviews: `{reviews}`. Reflect on this history and the accompanying reviews to infer a general type or mood of books they might be interested in next, without narrowing down to a specific genre or characteristic. `{constraint}`
(11)	Given the user's recent reading history: `{interaction}` and their reviews: `{reviews}`, suggest a broad intention for what they may want to read next, focusing on an overall style or feeling rather than pinpointing a particular book or specific genre. `{constraint}`
(12)	Based on these books: `{interaction}` and the corresponding reviews: `{reviews}`, generate an open-ended intention that represents a general mood or broad category the user could be leaning towards, even if their specific preferences aren't clear. `{constraint}`
(13)	The user has read these books: `{interaction}` and provided the following reviews: `{reviews}`. Use this data to determine a specific book intention they might have, such as seeking a particular genre, a specific plot, or a book with certain defining characteristics. `{constraint}`
(14)	Based on the user's recent book list: `{interaction}` and their reviews: `{reviews}`, infer a clearly defined intention for the next type of book they may want to read, focusing on particular elements like genre, theme, or distinctive features. `{constraint}`
(15)	Considering the user's reading pattern: `{interaction}` and their reviews: `{reviews}`, determine a specific intention about the next book they are likely to read, including precise details about the genre, mood, or main elements they are interested in. `{constraint}`
(16)	The user has recently read these books: `{interaction}` and left the following reviews: `{reviews}`. Based on this history, suggest an exploratory intention where the user might want to explore genres or types of books they haven't typically read. `{constraint}`
(17)	Given the user's book-reading history of: `{interaction}` and their reviews: `{reviews}`, infer an exploratory intention indicating their curiosity to explore new and different genres, styles, or narrative types that they might not have considered before. `{constraint}`
(18)	Using the following interaction data: `{interaction}` and corresponding reviews: `{reviews}`, generate an exploratory intention for the user, where they express interest in trying out new genres, themes, or book types that differ from their usual choices. `{constraint}`

Table 22: Generation templates for the Books dataset

	Templates of Generation of Engagement Profile (Movies & TV)
(1)	Analyze the user’s viewing history: `{interaction}` and the associated reviews: `{reviews}`. From these data points, determine the user’s explicit preferences, such as the genres, themes, or specific movie characteristics they explicitly praise or mention in their comments. `{constraint}`
(2)	Considering the user’s viewing history of these movies/TV shows: `{interaction}`, along with their corresponding reviews: `{reviews}`, generate a description of the user’s implicit preferences, focusing on any recurring genres, themes, or patterns evident in their comments. `{constraint}`
(3)	Based on the user’s recent engagement with the following movies/TV shows: `{interaction}` and their comments: `{reviews}`, identify their implicit preferences by analyzing the sentiments and focus of their reviews, such as preferred genres, themes, or character types. `{constraint}`
(4)	Analyze the user’s viewing history: `{interaction}` and the associated reviews: `{reviews}`. From these data points, determine the user’s explicit preferences, such as the genres, themes, or specific movie characteristics they explicitly praise or mention in their comments. `{constraint}`
(5)	The user has shown a clear interest in certain movies/TV shows: `{interaction}`, with specific comments: `{reviews}`. Using this data, infer their explicit preferences, such as favorite themes, plot types, or emotional tones they often highlight in their reviews. `{constraint}`
(6)	Consider the user’s engagement with these movies/TV shows: `{interaction}`, accompanied by their reviews: `{reviews}`. Based on these reviews, identify explicit preferences, such as preferred directors, frequent actors, or narrative styles that the user frequently praises or critiques. `{constraint}`
(7)	The user has recently reviewed a variety of different genres or unconventional movies/TV shows: `{interaction}`, with comments: `{reviews}`. Describe the user’s novelty preferences, such as their openness to experimenting with new genres or exploring unique cinematic styles, based on the diversity of their reviews. `{constraint}`
(8)	Given the user’s diverse viewing history: `{interaction}` and their reviews: `{reviews}`, identify any novelty preferences they may have, such as a tendency to seek out unique film experiences or genres that are outside their usual interests. `{constraint}`
(9)	The user has moved from watching typical genres to exploring less common ones: `{interaction}`, as indicated by their reviews: `{reviews}`. Describe the user’s novelty preferences, focusing on their interest in discovering new genres or unconventional storytelling approaches. `{constraint}`
(10)	The user has recently watched the following movies: `{interaction}`, with the following reviews: `{reviews}`. Reflect on this history and the accompanying reviews to infer a general type or mood of movies they might be interested in next, without narrowing down to a specific genre or characteristic. `{constraint}`
(11)	Given the user’s recent viewing history: `{interaction}` and their reviews: `{reviews}`, suggest a broad intention for what they may want to watch next, focusing on an overall style or feeling rather than pinpointing a particular movie or specific genre. `{constraint}`
(12)	Based on these movies: `{interaction}` and the corresponding reviews: `{reviews}`, generate an open-ended intention that represents a general mood or broad category the user could be leaning towards, even if their specific preferences aren’t clear. `{constraint}`
(13)	The user has watched these movies: `{interaction}` and provided the following reviews: `{reviews}`. Use this data to determine a specific movie intention they might have, such as seeking a particular genre, a specific plot, or a film with certain defining characteristics. `{constraint}`
(14)	Based on the user’s recent movie list: `{interaction}` and their reviews: `{reviews}`, infer a clearly defined intention for the next type of film they may want to watch, focusing on particular elements like genre, theme, or distinctive features. `{constraint}`
(15)	Considering the user’s viewing pattern: `{interaction}` and their reviews: `{reviews}`, determine a specific intention about the next movie they are likely to watch, including precise details about the genre, mood, or main elements they are interested in. `{constraint}`
(16)	The user has recently watched these movies: `{interaction}` and left the following reviews: `{reviews}`. Based on this history, suggest an exploratory intention where the user might want to explore genres or types of movies they haven’t typically watched. `{constraint}`
(17)	Given the user’s movie-watching history of: `{interaction}` and their reviews: `{reviews}`, infer an exploratory intention indicating their curiosity to explore new and different genres, styles, or narrative types that they might not have considered before. `{constraint}`
(18)	Using the following interaction data: `{interaction}` and corresponding reviews: `{reviews}`, generate an exploratory intention for the user, where they express interest in trying out new genres, themes, or movie types that differ from their usual choices. `{constraint}`

Table 23: Generation templates for the Movies & TV datasets



Given the user’s historical interactions, generate a personalized recommendation based on the user's engagement profile. If candidates are provided, use them; otherwise, create a new recommendation.

User's History: [historical interactions].
Engagement Profile: [Engagement Profile].
Candidates:

- **Generative Recommendation:** [NULL].
- **Direct Recommendation:** [candidate items].
- **Reranking:** [candidate items].
- **Matching:** [single candidate item].

[next item/reranked list/binary answer].

Figure 5: Recommendation Task.